IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:)	
)	
Darlet)	
)	
Serial No.: 09/754,785)	Group Art Unit: 2192
)	
Filed: January 4, 2001)	Examiner: Eric B. Kiss
)	
)	**Board of Patent Appeals and**
For: SYSTEM AND METHOD FOR)	**Interferences**
LINEAR PROCESSING OF)	
SOFTWARE MODULES)	
)	
Confirmation No.: 3238)	

Mail Stop: Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

In support of the Notice of Appeal filed on March 30, 2009, and pursuant to 37 C.F.R. § 41.37, Appellant presents this Appeal Brief in the above-captioned application.

This is an appeal to the Board of Patent Appeals and Interferences from the Examiner's final rejection of claims 1-15 and 40-60 in the Final Office Action dated December 31, 2008. The appealed claims are set forth in the attached Claims Appendix.

1.   Real Party in Interest

This application is assigned to Wind River Systems, Inc., the real party in interest.

2.   Related Appeals and Interferences

There are no other appeals or interferences that would directly affect, be directly affected, or have a bearing on the instant appeal.

3.   Status of the Claims

Claims 1-15 and 40-60 have been rejected in the Final Office Action. Claims 16-39 were canceled in a previous amendment. The final rejection of claims 1-15 and 40-60 is being appealed.

4.   Status of Amendments

All amendments submitted by Appellant have been entered.

5.   Summary of Claimed Subject Matter

The present invention, as recited in independent claim 1, relates to a computer readable storage medium having a set of instructions that, when executed, is operable to receive a software module, wherein the software module include references to locations within the software module and at least some of the references are backward references. (See Specification, p. 5, ll. 16-19). The method also includes reordering components of the software module to remove at least some of the backward references, with the reordering done based on a type of the components. (See Id., at p. 5, ll. 25-29; p. 32, ll. 7-15). The components include at least one of a header, a section, and a table. (See Id., at p. 6, ll. 27-28). The reordered software module includes the at least some of the backward references, and the at least some of the backward

references in the reordered software module are stored in a memory to avoid a nonsequential reading of the reordered software module. (See Id., at p. 12, ll. 20-27).

The present invention, as recited in independent claim 9, is directed to a system that comprises a reorder module. (See Id., p. 13, l. 32 - p. 14, l. 4). The reorder module is configured to receive a software module including references to locations within the software module, wherein at least some of the references are backward references. (See Id.) In addition, the reorder module is configured to reorder components of the software module and remove at least some of the backward references, with the reordering done based on a type of the components . (See Id. and p. 32, ll. 7-15). The components include at least one of a header, a section, and a table. (See Id., at p. 6, ll. 27-28). The system also includes a processor executing the reorder module, wherein the reordered software module includes the at least some of the backward references, and wherein the at least some of the backward references in the reordered software module are stored in a memory to avoid a nonsequential reading of the reordered software module. (See Id., at page 11, l. 27, to p. 12, l. 5).

The present invention, as recited in independent claim 55, is directed to a computer readable storage medium provided with a set of instructions that, when executed, is operable to receiving a software module that includes components arranged in a first order. (See Id., at p. 10, l. 28 - p. 11, l. 8). In addition, a first one of the components includes a reference to a location in a second component, wherein the second component precedes the first component in the first order. (See Id.). The method further includes arranging the components into a second order so that the second component is subsequent to the first component in the second order, with the arrangement being based on a type of the first and second components. (See Id. and p. 32, ll. 7-15). The components include at least one of a header, a section, and a table. (See Id., at p. 6, ll. 27-28). The reordered software module includes at least one reference from a third component to a preceding component, and the at least one reference from the third component is stored in a memory to avoid a nonsequential reading of the reordered software module. (See Id., at page 11, l. 27, to p. 12, l. 5).

3

6.   Ground of Rejection to be Reviewed on Appeal

I.   Whether claims 1-15, 40, 41, and 43-60 are unpatentable over 35 U.S.C. §
103(a) by "Linkers and Loaders, chapter 6" by John Levine, June 1999
(hereinafter "Levine").

II.   Whether claim 42 is unpatentable over 35 U.S.C. § 103(a) by Levine in
view of U.S. Patent No. 6,185,733 to Breslau et al. (hereinafter "Breslau").

7.   Argument

I.   Claims 1-15, 40, 41, and 43-60 should not be rejected under 35 U.S.C. §
103(a) based on "Linkers and Loaders, chapter 6" by Levine.

Claim 1 recites "[a] computer readable storage medium including a set of instructions
executable by a processor, the set of instructions operable to: receive a software module, the
software module including references to locations within the software module, at least some of
the references being backward references; and reorder components of the software module into a
predetermined order based on a type of the components to remove at least some of the backward
references, wherein the components include at least one of a header, a section, and a table,
wherein the reordered software module includes the at least some of the backward references,
and wherein the at least some of the backward references in the reordered software module are
stored in a memory to avoid a nonsequential reading of the reordered software module."

The Examiner asserts that "*Levine* discloses... reordering components of the software
module into a predetermined order based on a type (*i.e.* objects being referenced) of the
components to remove at least some of the backward references." (12/31/08 Office Action, p. 3,
citing Levine, pp. 5-6.) The Appellant respectfully disagrees. Claim 1 recites that components
are reordered "into a *predetermined order* based on a *type* of the components." For example, in
the exemplary embodiment of Figures 1-4 of the Specification of the present invention, the
predetermined order recited includes a module header 310, a program header table 320, a section

4

header table 330, a section string table 340, an entry point table 350, a text section 360, one or more data sections 370, a symbol string table 380, a symbol table 390, and a relocation information table 395. (See Specification, pp. 5-17, Figs. 1-4.) While other orders may be possible in other exemplary embodiments, one of skill in the art will understand that this order is clearly predetermined and based on the functions of various components, as recited in claim 1. In contrast, Levine describes a method wherein the obtained order is dynamically determined based on symbol references contained within the various components. (See Levine, pp. 5-6, § "Creating libraries".) The Appellant respectfully submits that no "predetermined order" is disclosed or suggested by Levine, and that the ordering disclosed by Levine is not "based on a type of the components," and, therefore, Levine does not disclose or suggest "reorder[ing] components of the software module into a predetermined order based on a type of the components," as recited in claim 1. Accordingly, this rejection should be reversed. Because claims 2-8 and 40, 41 and 43-54 depend from, and, therefore, include all of the limitations of claim 1, it is respectfully submitted that these claims are also allowable for at least the foregoing reasons.

Claim 9 recites "[a] system, comprising: a memory storing a reorder module configured to receive a software module including references to locations within the software module, at least some of the references being backward references, the reorder module configured to reorder components of the software module into a predetermined order based on a type of the components and remove at least some of the backward references, the components including at least one of at least one of a header, a section, and a table; and a processor executing the reorder module, wherein the reordered software module includes the at least some of the backward references, and wherein the at least some of the backward references in the reordered software module are stored in a memory to avoid a nonsequential reading of the reordered software module.

The Appellant respectfully submits that Levine does not disclose or suggest a "reorder module configured to reorder components of the software module into a predetermined order based on a type of the components," as recited in claim 9, for the reasons discussed above with reference to claim 1. Accordingly, this rejection should be reversed. Because claims 10-15

depend from, and, therefore, include all of the limitations of claim 9, it is respectfully submitted that these claims are also allowable for at least the foregoing reasons.

Claim 55 recites "[a] computer readable storage medium including a set of instructions executable by a processor, the set of instructions operable to: receive a software module, the software module including components arranged in a first order, a first one of the components including a reference to a location in a second one of the components, the second one of the components preceding the first one of the components in the first order; and arrange the components into a predetermined second order to produce a reordered software module so that the second one of the components is subsequent to the first one of the components in the second order, wherein the arrangement is based on a type of the first and second ones of the components, wherein the components include at least one of a header, a section, and a table, wherein the reordered software module includes at least one reference from a third component to a preceding component, and wherein the at least one reference from the third component is stored in a memory to avoid a nonsequential reading of the reordered software module."

The Appellant respectfully submits that Levine does not disclose or suggest a "instructions operable to... arrange the components into a predetermined second order to produce a reordered software module so that the second one of the components is subsequent to the first one of the components in the second order, wherein the arrangement is based on a type of the first and second ones of the components," as recited in claim 55, for the reasons discussed above with reference to claim 1. Accordingly, this rejection should be reversed. Because claims 56-60 depend from, and, therefore, include all of the limitations of claim 55, it is respectfully submitted that these claims are also allowable for at least the foregoing reasons.

II.   Claim 42 should not be rejected under 35 U.S.C. § 103(a) based on Levine
        in view of U.S. Patent No. 6,185,733 to Breslau.

Claim 42 stands rejected under 35 U.S.C. § 103(a) as unpatentable over Levine as applied to claim 1, and further in view of U.S. Patent No. 6,185,733 to Breslau. (See 12/31/08 Office Action, p. 9.) The Appellant respectfully submits that Breslau does not cure the deficiencies of

Levine discussed above with reference to claim 1. Accordingly, this rejection should be reversed.

8.   Conclusion

For the reasons set forth above, Appellant respectfully requests that the Board reverse the above-discussed rejections of claims 1-15 and 40-60.

Respectfully submitted,

Date: 7/9/09

By: _____
Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, NY 10038
Tel.:   (212) 619-6000
Fax: (212) 619-0276

CLAIMS APPENDIX

1.   (Previously Presented) A computer readable storage medium including a set of instructions executable by a processor, the set of instructions operable to:

receive a software module, the software module including references to locations within the software module, at least some of the references being backward references; and

reorder components of the software module into a predetermined order based on a type of the components to remove at least some of the backward references,

wherein the components include at least one of a header, a section, and a table,

wherein the reordered software module includes the at least some of the backward references, and

wherein the at least some of the backward references in the reordered software module are stored in a memory to avoid a nonsequential reading of the reordered software module.

2.   (Previously Presented) The computer readable storage medium according to claim 1, wherein the set of instructions is further operable to:

adjust at least one of the references in the software module to reflect the reordering of the components of the software module, so that the at least one of the references remains a reference to the same component, but to the component's new, reordered location, the new, reordered location coming after the at least one reference in the software module.

3.   (Previously Presented) The computer readable storage medium according to claim 2, wherein the software module includes a symbol table, the symbol table including backward references when the reordering of the components of the software module and adjusting the at least one of the references have been completed.

4.   (Previously Presented) The computer readable storage medium according to claim 2, wherein the software module includes a symbol table, the software module including no backward references in locations before the symbol table when the reordering of the components of the software module and adjusting the at least one of the references have been completed.

5.   (Previously Presented) The computer readable storage medium according to claim 2, wherein the software module is a relocatable object code module in ELF format when the reordering the components of the software module and adjusting the at least one of the references have been completed.

6.   (Previously Presented) The computer readable storage medium according to claim 5, wherein, when the software module is received, the software module is a relocatable object code module in ELF format, and wherein, when the reordering the components of the software module and adjusting the at least one of the references have been completed, the software module includes a symbol table, the symbol table including backward references, and the software module includes no backward references from locations before the symbol table.

7.   (Previously Presented) The computer readable storage medium according to claim 1, wherein the software module comprises at least one segment, each at least one segment comprising at least one section, and wherein sections in the same segment are contiguously located in the software module when the reordering of the components of the software module has been completed.

8.   (Previously Presented) The computer readable storage medium according to claim 1, wherein, when the software module is received, the software module is a relocatable object code module in ELF format.

9. (Previously Presented) A system, comprising:

        a memory storing a reorder module configured to receive a software module including references to locations within the software module, at least some of the references being backward references, the reorder module configured to reorder components of the software module into a predetermined order based on a type of the components and remove at least some of the backward references, the components including at least one of at least one of a header, a section, and a table; and

        a processor executing the reorder module, wherein the reordered software module includes the at least some of the backward references, and

10

wherein the at least some of the backward references in the reordered software module are stored in a memory to avoid a nonsequential reading of the reordered software module.

10.   (Previously Presented) The system according to claim 9, wherein the reorder module is configured to adjust a reference in the software module to reflect the reordering of the components of the module.

11.   (Original) The system according to claim 9, wherein the software module includes a symbol table, and wherein the reorder module is configured not to remove backward references from the symbol table.

12.   (Original) The system according to claim 9, wherein the software module includes a symbol table, and wherein the reorder module is configured to remove all backward references from locations before the symbol table in the reordered software module.

13.   (Original) The system according to claim 9, wherein the software module includes at least one segment, each of the at least one segments including at least one section, and the reorder module is configured to locate sections in the same segment contiguously in the reordered software module.

14.   (Original) The system according to claim 9, wherein the software module is a relocatable object code module in ELF format, and the reordered software module is a relocatable object code module in ELF format.

15.   (Previously Presented) The system according to claim 14, wherein the software module includes a symbol table, wherein the reorder module is configured to adjust a reference in the software module to reflect the reordering of the components of the module, wherein the reorder module is configured to remove all backward references from locations before the symbol table, and wherein the reorder module is configured not to remove backward references from the symbol table.

16. (Cancelled) .

17. (Cancelled).

18. (Cancelled).

19. (Cancelled).

20. (Cancelled).

21. (Cancelled).

22. (Cancelled)

23. (Cancelled).

24. (Cancelled).

25. (Cancelled).

26. (Cancelled).

27. (Cancelled).

28. (Cancelled).

29. (Cancelled).

30. (Cancelled).

31. (Cancelled).

32. (Cancelled).

33. (Cancelled).

34. (Cancelled).

35. (Cancelled).

36. (Cancelled).

37. (Cancelled).

38. (Cancelled).

39. (Cancelled).

40. (Previously Presented) The computer readable storage medium of claim 1, wherein the reordering of the components of the software module is completed prior to linking the software module.

41. (Previously Presented) The computer readable storage medium of claim 40, wherein the set of instructions is further operable to:
link the reordered software module.

42. (Previously Presented) The computer readable storage medium of claim 1, wherein the set of instructions is further operable to:
transfer the reordered software module to a different computer system; and linking the reordered software module on the different computer system.

43.   (Previously Presented) The computer readable storage medium of claim 1, wherein the reordered components include an ELF data section.

44.   (Previously Presented) The computer readable storage medium of claim 1, wherein the reordered components include an ELF code section.

45.   (Previously Presented) The computer readable storage medium of claim 1, wherein the reordered components include an ELF header table.

46.   (Previously Presented) The computer readable storage medium of claim 1, wherein the reordered components include an ELF entry point table.

47.   (Previously Presented) The computer readable storage medium of claim 1, wherein the reference points to a section located prior to the reference in the received software module.

48.   (Previously Presented) The computer readable storage medium of claim 47, wherein, after the software module has been reordered, the reference is changed to point at the same section, the section having been relocated to appear after the reference in the reordered software module.

49.   (Previously Presented) The computer readable storage medium of claim 1, wherein the reference points to a table located prior to the reference in the received software module.

50.   (Previously Presented) The computer readable storage medium of claim 49, wherein, after the software module has been reordered, the reference is changed to point at the same table, the table having been relocated to appear after the reference in the reordered software module.

51.   (Previously Presented) The computer readable storage medium of claim 1, wherein the reference points into a section located prior to the reference in the received software module.

52.   (Previously Presented) The computer readable storage medium of claim 51, wherein, after the software module has been reordered, the reference points into the same section, the section having been relocated to appear after the reference in the reordered software module.

53.   (Previously Presented) The computer readable storage medium of claim 1, wherein the reference points into a table located prior to the reference in the received software module.

54.   (Previously Presented) The computer readable storage medium of claim 53, wherein, after the software module has been reordered, the reference is changed to point into the same table, the table having been relocated to appear after the reference in the reordered software module.

55.   (Previously Presented) A computer readable storage medium including a set of instructions executable by a processor, the set of instructions operable to:

receive a software module, the software module including components arranged in a first order, a first one of the components including a reference to a location in a second one of the components, the second one of the components preceding the first one of the components in the first order; and

arrange the components into a predetermined second order to produce a reordered software module so that the second one of the components is subsequent to the first one of the components in the second order, wherein the arrangement is based on a type of the first and second ones of the components,

wherein the components include at least one of a header, a section, and a table,

wherein the reordered software module includes at least one reference from a third component to a preceding component, and

wherein the at least one reference from the third component is stored in a memory to avoid a nonsequential reading of the reordered software module.

56.   (Previously Presented) The computer readable storage medium of claim 55, wherein the arranging occurs prior to linking the software module.

15

57.   (Previously Presented) The computer readable storage medium of claim 56, wherein the set of instructions is further operable to:

      linking the software module without storing the entire software module in local memory.

58.   (Previously Presented) The computer readable storage medium of claim 57, wherein the components include an ELF table and an ELF section.

59.   (Previously Presented) The computer readable storage medium of claim 58, wherein the order of segments within the ELF section is preserved when the section is moved to a different position in the reordered software module.

60.   (Previously Presented) The computer readable storage medium of claim 59, wherein the only backward references between different ELF components in the reordered software module are references located in the ELF symbol table.

EVIDENCE APPENDIX

No evidence has been submitted herewith or is relied upon in the present appeal.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings or decisions which relate to the present appeal.